8

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER OF PATENTS AND TRADEMARKS
Washington, D.C. 20231
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/378,596	08/20/1999	SHAIL ADITYA GUPTA	HP10981866-1	9330

22879 7590 10/04/2002

HEWLETT PACKARD COMPANY
P O BOX 272400, 3404 E. HARMONY ROAD
INTELLECTUAL PROPERTY ADMINISTRATION
FORT COLLINS, CO 80527-2400

EXAMINER	
MAKHDOOM, SAMARINA	
ART UNIT	**PAPER NUMBER**
2123	

DATE MAILED: 10/04/2002

Please find below and/or attached an Office communication concerning this application or proceeding.

PTO-90C (Rev. 07-01)

Office Action Summary	Application No. 09/378,596	Applicant(s) GUPTA ET AL.
	Examiner Samarina Makhdoom	Art Unit 2123

-- *The MAILING DATE of this communication appears on the cover sheet with the correspondence address* --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133).
- Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *20 August 1999* .

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-18* is/are pending in the application.

4a) Of the above claim(s) ______ is/are withdrawn from consideration.

5)☐ Claim(s) ______ is/are allowed.

6)☒ Claim(s) *1-18* is/are rejected.

7)☐ Claim(s) ______ is/are objected to.

8)☐ Claim(s) ______ are subject to restriction and/or election requirement.

Application Papers

9)☒ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on ______ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

11)☐ The proposed drawing correction filed on ______ is: a)☐ approved b)☐ disapproved by the Examiner.

If approved, corrected drawings are required in reply to this Office action.

12)☐ The oath or declaration is objected to by the Examiner.

Priority under 35 U.S.C. §§ 119 and 120

13)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. ______ .

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

14)☐ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. § 119(e) (to a provisional application).

a) ☐ The translation of the foreign language provisional application has been received.

15)☐ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. §§ 120 and/or 121.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☒ Information Disclosure Statement(s) (PTO-1449) Paper No(s) *4 & 6* .
4) ☐ Interview Summary (PTO-413) Paper No(s). ______ .
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other:

DETAILED ACTION

Specification

1. The specification is objected to because of minor informalities. The header "HP10981866-1 22 Express Mail No. EL121361078US" on all pages of the specification should be removed. Correction is required.

Claim Interpretation

2. The term 'functional unit' is defined as unit responsible for executing operations supported in the processor's instruction format. See applicant's specification, page 1. Therefore, with this definition a functional unit is interpreted to be an adder unit, a multiplier unit, or an ALU.

Claim Rejections - 35 USC § 102

3. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.
>
> (e) the invention was described in a patent granted on an application for patent by another filed in the United States before the invention thereof by the applicant for patent, or on an international application by another who has fulfilled the requirements of paragraphs (1), (2), and (4) of section 371(c) of this title before the invention thereof by the applicant for patent.

The changes made to 35 U.S.C. 102(e) by the American Inventors Protection Act of 1999 (AIPA) do not apply to the examination of this application as the application being examined was not (1) filed on or after November 29, 2000, or (2) voluntarily published under 35 U.S.C.

122(b). Therefore, this application is examined under 35 U.S.C. 102(e) prior to the amendment by the AIPA (pre-AIPA 35 U.S.C. 102(e)).

4. Claims 1-18 are rejected under 35 U.S.C. 102(b) as being anticipated by Dey et al. U.S. Patent No. 5,513,123.

As per Claims 1 and 9, Dey et al. disclose a method for automatic design of a processor datapath from an input specification including a register file specification, a set of specified processor operations and a desired instruction level parallelism among the specified operations, the method comprising:

determining sets of mutually exclusive operations from the specified processor operations based on the desired instruction level parallelism (See Figure 1a, and text on Col. 6 lines 14-50, the figure displays the parallelism of processor operations through the adder and multiplier units);

programmatically assigning instances of functional units from a macrocell library to the sets of mutually exclusive operations, such that each specified operation is associated with a corresponding functional unit (See Figure 1, and text on Col. 6 lines 14-50 on characteristics of the data paths. The text discloses the specified operations associated with of adders, multipliers, or functional units);

programmatically determining a resource allocation of register file ports to ports of the functional units; and programmatically synthesizing register files with the allocated read/write ports and interconnects between the functional units and the allocated read/write ports (See Col. 9, lines 55-65 for register files receiving data and sending data. The act of receiving and sending data is equivalent to the read/write function, and read/write ports are inherent to a register file).

As per Claim 2, Dey et al. disclose the ports of the functional units each have a corresponding register file port request and programmatically determining the resource allocation includes: programmatically allocating a minimum number of read/write ports that satisfies all of the port requests (See Col. 4, lines 20-25 discuss using an approach that minimizes hardware to view all datapath loops. Also See Col. 9, lines 55-65 for register files that receive and send data from a particular Execution unit (or ALU or functional unit) therefore programmatically allocating resources).

As per Claim 3, Dey et al. disclose the specification of parallelism among operations is specified as exclusion relationships among operations that indicate which operations cannot be executed concurrently (See Col. 3, lines 36-46, discussing the execution units and the dependencies between the execution units. These dependencies will determine what operations can execute concurrently).

As per Claim 4, Dey et al. disclose the input specification further includes: a mapping between the specified operations and register file types in the register file specification; and operation formats describing inputs and outputs of the specified operations (See Col. 9. lines 41-54 to see organization of register files and the disclosure of controllability points to the register files so they are associated with a particular execution unit or functional unit).

As per Claim 5, Dey et al. disclose the synthesized functional units include macrocell instances, the synthesized register files include register file instances, and the interconnect includes macrocell instances of wires, buses, muxes, or tri-states (See Figure 1a, and corresponding text Col. 6, lines 20-29 showing interconnects of macrocell instrances such as buses, wires, adders, and multipliers).

As per Claim 6, Dey et al. disclose determining sets of mutually exclusive operations includes: finding maximal cliques of mutually exclusive operations based on exclusion relations derived from the input specification (See Col. 9, lines 58-65 for the used of cliques based on the self-looping registers. The larger the number of self-looping registers, the larger the clique).

As per Claim 7, Dey et al. disclose synthesizing functional units includes: building a list of valid functional units based on opcodes and latency of the specified operations (See Figure 1a. and text on Col. 6, lines 14-50, a list of valid functions are displayed in this figure and Table I);

from the list, selecting functional units such that each functional unit covers a maximum number of operations in a set of mutually exclusive operations (See Figure 1a, and corresponding text Col. 6, lines 20-29 showing interconnects i.e. buses, adders, and multipliers; the connections shown on Figure 1a allow to the system to perform different operations simultaneously covering the maximum number of operations).

As per Claim 8, Dey et al. disclose using the instruction level parallelism from the input specification to identify which functional unit ports can be allocated to the same register port, and allocating selected functional unit ports to a single, shared register port (See Col. 9, lines 31-54 showing the different inputs and outputs (or ports) of the register files and the use of test pins as test 'ports'. Furthermore, the output of functional unit A2 is multiplexed with the test points ntest to ensure the functionality of the datapath remains unchanged. The same register file ports of RA2 and LA2 are associated with the functional unit port of A2).

As per Claims 10 and 15, Dey et al. disclose a method for automatic synthesis of functional units in a programmable processor datapath, the method comprising:

from an input specification defining a set of specified processor operations and instruction level parallelism among the specified operations, determining sets of mutually exclusive operations (See Figure 1a, and text in Col. 6 lines 14-50 for the parallelism of the adder and multiplier operations);

programmatically assigning instances of functional units from a macrocell library to the sets of mutually exclusive operations, such that each specified operation is associated with a corresponding functional unit (See Figure 1a, and text in Col. 6 lines 14-50 for the parallelism of the adder and multiplier operations, the add function is associated with the adder functional unit and the multiply function is associated with the multiply functional unit);

and programmatically synthesizing the functional units from the macrocell library such that the functional units are described in a hardware description language (See Col. 2, lines 44-66 for the used of hardware description languages such as RTL to define and test a circuit).

As per Claim 11, Dey et al. disclose determining sets of mutually exclusive operations includes: finding exclusion cliques where each clique represents a maximal set of mutually exclusive operations; and wherein assigning instances of functional units includes programmatically selecting instances of functional units to cover the cliques from the macrocell library (See Col. 9, lines 55 to 62 for the use of register cliques, each set of registers such as RA2 or RA1 as assigned to a functional unit).

As per Claim 12, Dey et al. disclose synthesizing functional units includes: building a list of valid functional units based on opcodes of the specified operations; from the list, selecting functional units such that each functional unit covers a maximum number of operations in a set of mutually exclusive operations (See Col. 15, lines 63-67, where Dey et al. disclose

maximizing control and observation points for the datapath, this same method could be applied to datapath functional units and opcodes).

As per Claim 13, Dey et al. disclose functional unit instances are assigned such that the semiconductor area covered by functional units in the processor design is minimized (The design goal to minimize semiconductor area is inherent to the semiconductor art).

As per Claim 14, Dey et al. disclose the functional unit instances are assigned such that the number of operations covered by each of the functional unit instances is maximized (See Col. 15, lines 63-67, where Dey et al. disclose maximizing control and observation points for the datapath, this same method could be applied to datapath functional units, parallelism, and opcodes).

As per Claim 16 and 18, Dey et al. disclose a method for automatic synthesis of a register file and functional unit-register file interconnect in a processor, based on an input specification of register file types in the processor, specified processor operations, desired instruction level parallelism among the specified operations and functional units in the processor, the method comprising:

for each type of register file specified in the processor, establishing a set of read/write port requests between the functional units and each of the register file types (See Col. 4, lines 20-25 discuss using an approach that minimizes hardware to view all datapath loops. Also See Col. 9, lines 55-65 for register files that receive and send data from a particular Execution unit (or ALU or functional unit) therefore programmatically allocating resources);

programmatically computing a resource allocation of register ports in the register file types to read/write port requests, including determining how to share a register port for two or

more functional unit ports based on the specification of instruction level parallelism among the operations and (See Col. 4, lines 20-25 discuss using an approach that minimizes hardware to view all datapath loops. Also See Col. 9, lines 55-65 for register files that receive and send data from a particular Execution unit (or ALU or functional unit) therefore programmatically allocating resources);

programmatically synthesizing register files with the allocated read/write ports and interconnects between the functional units and the allocated read/write ports. (See Col. 9, lines 55-65 for register files receiving data and sending data The sending and receiving data function are equivalent to the read/write function and an apparatus for reading/writing such as ports are inherent to a register file).

As per Claim 17, Dey et al. disclose the resource allocation uses a contiguous allocation heuristic that simplifies interconnect layout by allocating register port requests from a functional unit to contiguous register ports (See Col. 6, lines 14-30 for datapath layout and Figure 1a for register port and functional unit connections).

5. Claims 1-5, 7-10, and 12-18 are rejected under 35 U.S.C. 102(e) as being anticipated by Abbott U.S. Patent No. 6,351,142.

As per Claims 1 and 9, Abbott discloses a method for automatic design of a processor datapath from an input specification including a register file specification, a set of specified processor operations and a desired instruction level parallelism among the specified operations, the method comprising:

determining sets of mutually exclusive operations from the specified processor operations based on the desired instruction level parallelism (See Col. 5, lines 5-19 disclose how each datapath is configured to perform logic and arithmetic operations);

programmatically assigning instances of functional units from a macrocell library to the sets of mutually exclusive operations, such that each specified operation is associated with a corresponding functional unit (See Col. 6, lines 11-19 to see how opcodes can be in a microprocessor coupled to a FPGA to control operations. Part of controlling operations is assigning resources to specific operations).

programmatically determining a resource allocation of register file ports to ports of the functional units; and programmatically synthesizing register files with the allocated read/write ports and interconnects between the functional units and the allocated read/write ports (See Col. 4, line 61 to Col. 5 line 10, disclosing how the register files are dual ported for read operations and single ported for write operations. A selector is also disclosed for selecting registers for specific functions).

As per Claim 2, Abbott discloses the ports of the functional units each have a corresponding register file port request and programmatically determining the resource allocation includes: programmatically allocating a minimum number of read/write ports that satisfies all of the port requests (See Col. 4, line 61 to Col. 5 line 10, disclosing how the register files are dual ported for read operations and single ported for write operations. A selector is also disclosed for selecting registers for specific functions).

As per Claim 3, Abbott discloses the specification of parallelism among operations is specified as exclusion relationships among operations that indicate which operations cannot be

executed concurrently (See Col. 3, lines 20-26, for concurrent control of logic operations. The control bits determine what operations cannot be executed concurrently and provide dynamic programmability to logic operations on inputs and outputs).

As per Claim 4, Abbott discloses the input specification further includes: a mapping between the specified operations and register file types in the register file specification; and operation formats describing inputs and outputs of the specified operations (See Col. 3, lines 20-26, for concurrent control of logic operations. The control bits determine what operations cannot be executed concurrently and provide dynamic programmability to logic operations on inputs and outputs).

As per Claim 5, Abbott discloses the synthesized functional units include macrocell instances, the synthesized register files include register file instances, and the interconnect includes macrocell instances of wires, buses, muxes, or tri-states (See Col. 19, lines 40-49 for discussion of buses used in data transfer, See col. 15, lines 28-35 for muxes used in logic operations).

As per Claim 7, Abbott discloses synthesizing functional units includes: building a list of valid functional units based on opcodes and latency of the specified operations; from the list, selecting functional units such that each functional unit covers a maximum number of operations in a set of mutually exclusive operations (Col. 1, lines 38-52, for instruction bits and the control or 'opcodes' of datapath operations. The functional unit or ALU can execute either arithmetic and/or Boolean operations as realized by the datapath of the CPU).

As per Claim 8, Abbott discloses using the instruction level parallelism from the input specification to identify which functional unit ports can be allocated to the same register port,

and allocating selected functional unit ports to a single, shared register port (See Col. 4, line 61 to Col. 5 line 10, disclosing how the register files are dual ported for read operations and single ported for write operations. A selector is also disclosed for selecting registers for specific functions).

As per Claims 10 and 15, Abbott discloses a method for automatic synthesis of functional units in a programmable processor datapath, the method comprising: from an input specification defining a set of specified processor operations and instruction level parallelism among the specified operations, determining sets of mutually exclusive operations (See Figure1, for two register banks in parallel and a selector unit for selecting the specified processor operation. The operation is executed in parallel by either of the datapath units connected to the output of the selector circuitry. Col. 6, lines 5-19, for the disclosure of parallel hardware to switch control threads on a cycle-by-cycle basis);

programmatically assigning instances of functional units from a macrocell library to the sets of mutually exclusive operations, such that each specified operation is associated with a corresponding functional unit (See Figure 1, and Col 5, lines 24 et Seq for the selector unit assigning operations to either the arithmetic or logic datapath depending on the specified operation);

and programmatically synthesizing the functional units from the macrocell library such that the functional units are described in a hardware description language (Col. 9, lines 22-34, for the disclosure of HDLs such as logic synthesis tools).

As per Claim 12, Abbott discloses synthesizing functional units includes: building a list of valid functional units based on opcodes of the specified operations (See Col. 2, line 30 et seq

for the disclosure of defining a highly specialized datapath that implements several specialized instructions supplied by the CPU. The instructions supplied by the CPU are in list format with opcodes and specified operations for the datatpath);

from the list, selecting functional units such that each functional unit covers a maximum number of operations in a set of mutually exclusive operations (See Col. 6, lines 42 et Seq for the disclosure of rearrangement. The rearrangement of bits can include output from a previous logic or arithmetic function. Therefore, the input, control, and output bits can be rearranged for the maximum number of operations for this configuration).

As per Claim 13, Abbott discloses functional unit instances are assigned such that the semiconductor area covered by functional units in the processor design is minimized (The design goal to minimize semiconductor area of all elements so that most efficient space allocation is achieved is inherent to the semiconductor art).

As per Claim 14, Abbott discloses the functional unit instances are assigned such that the number of operations covered by each of the functional unit instances is maximized (See Col. 17, lines 5 et Seq. for the disclosure of dynamic programming to adjust the configuration to a particular application. The motivation to modify a the disclosed invention per application is to maximize the use of the functional units in the invention; and See Col. 6, lines 42 et Seq for the disclosure of rearrangement. The rearrangement of bits can include output from a previous logic or arithmetic function. Therefore, the input, control, and output bits can be rearranged for the maximum number of operations for this configuration).

As per Claim 16 and 18, Abbott discloses a method for automatic synthesis of a register file and functional unit-register file interconnect in a processor, based on an input specification

of register file types in the processor, specified processor operations, desired instruction level parallelism among the specified operations and functional units in the processor, the method comprising:

for each type of register file specified in the processor, establishing a set of read/write port requests between the functional units and each of the register file types (See Col. 4, line 61 to Col. 5 line 10, disclosing how the register files are dual ported for read operations and single ported for write operations. A selector is also disclosed for selecting registers for specific functions);

programmatically computing a resource allocation of register ports in the register file types to read/write port requests, including determining how to share a register port for two or more functional unit ports based on the specification of instruction level parallelism among the operations (See Col. 4, line 61 to Col. 5 line 10, disclosing how the register files are dual ported for read operations and single ported for write operations. A selector is also disclosed for selecting registers for specific functions); and

programmatically synthesizing register files with the allocated read/write ports and interconnects between the functional units and the allocated read/write ports (See Col. 4, line 61 to Col. 5 line 10, disclosing how the register files are dual ported for read operations and single ported for write operations. A selector is also disclosed for selecting registers for specific functions).

As per Claim 17, Abbott discloses the resource allocation uses a contiguous allocation heuristic that simplifies interconnect layout by allocating register port requests from a functional unit to contiguous register ports (See Col. 15, lines 10 et Seq for the discussion of pins i.e. ports

and their arrangement also See Figure 2, for the rearrangement circuit to rearrange the register ports of the datapath for optimal application execution).

Conclusion

6. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

Miller et al. U.S. Patent No. 5,831,991 disclose an apparatus for electrically verifying a functional unit contained within an integrated circuit.

Guerra et al. U.S. Patent No. 5,502,645 disclose a method to compose a reconfigurable built-in-self-repair for a datapath.

Sprague et al. U.S. Patent No. 5,530,884 disclose a method for processing data for an apparatus comprising of a plurality of datapaths.

7. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Samarina Makhdoom whose telephone number is 703-305-7209. The examiner can normally be reached on Full Time on Tuesday, Thursday, Friday, and Sunday.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Kevin J. Teska can be reached on 703-305-9704. The fax phone numbers for the organization where this application or proceeding is assigned are 703-305-0040 for regular communications and 703-305-0040 for After Final communications.

Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the receptionist whose telephone number is 703-305-3900.

SM
October 1, 2002

KEVIN J. TESKA
SUPERVISORY
PATENT EXAMINER